Bayesian Methods in CS&E Models

Laura Swiler

Sandia National Laboratories
Optimization and Uncertainty Estimation Dept., MS 0370
Albuquerque, NM 87185
Email: lpswile@sandia.gov

SIAM Conference
Feb. 15, 2005
SAND 2005-0463C

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Bayesian Analysis

- **Construct a prior distribution on a parameter (which might be a parameter of a distribution)**
- **The prior distribution should be based on previous experience, engineering judgment**
- **The distribution on the prior is updated with actual data. The resulting updated distribution is called the posterior.**

Frequentist	Bayesian
Assumes there is an unknown but fixed parameter θ	Assumes a distribution on unknown parameter θ
Estimates θ with some confidence interval	Uses probability theory, treats θ as a random variable

Bayesian Analysis

- **Why would we use it for CS&E problems?**
- **Nice feature of incorporating additional data as it becomes available**
- **We often don’t have good estimates: Bayes provides a framework for starting with what we do know, and refining our estimates in a statistically consistent manner**
- **Examples:**
  - Reliability problems: Update probability of failure
  - Response surfaces: Update parameters in a surrogate model for a trust region
  - Calibration under Uncertainty (CUU): Update our parameter estimates based on experimental data AND uncertainty in a model

Bayesian Methods

Discrete Case

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x},\theta)}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid \theta)\,p(\theta)}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid \theta)\,p(\theta)}{\sum_{\theta} p(\mathbf{x} \mid \theta)\,p(\theta)}$$

where θ is a parameter(s), x is a data vector, and p is a probability mass function.

$$p(\theta \mid \mathbf{x}) = posterior \propto p(\mathbf{x} \mid \theta)\,p(\theta) = likelihood * prior$$

Examples

- Use Binomial distribution to model the number of failures, x, in n trials.

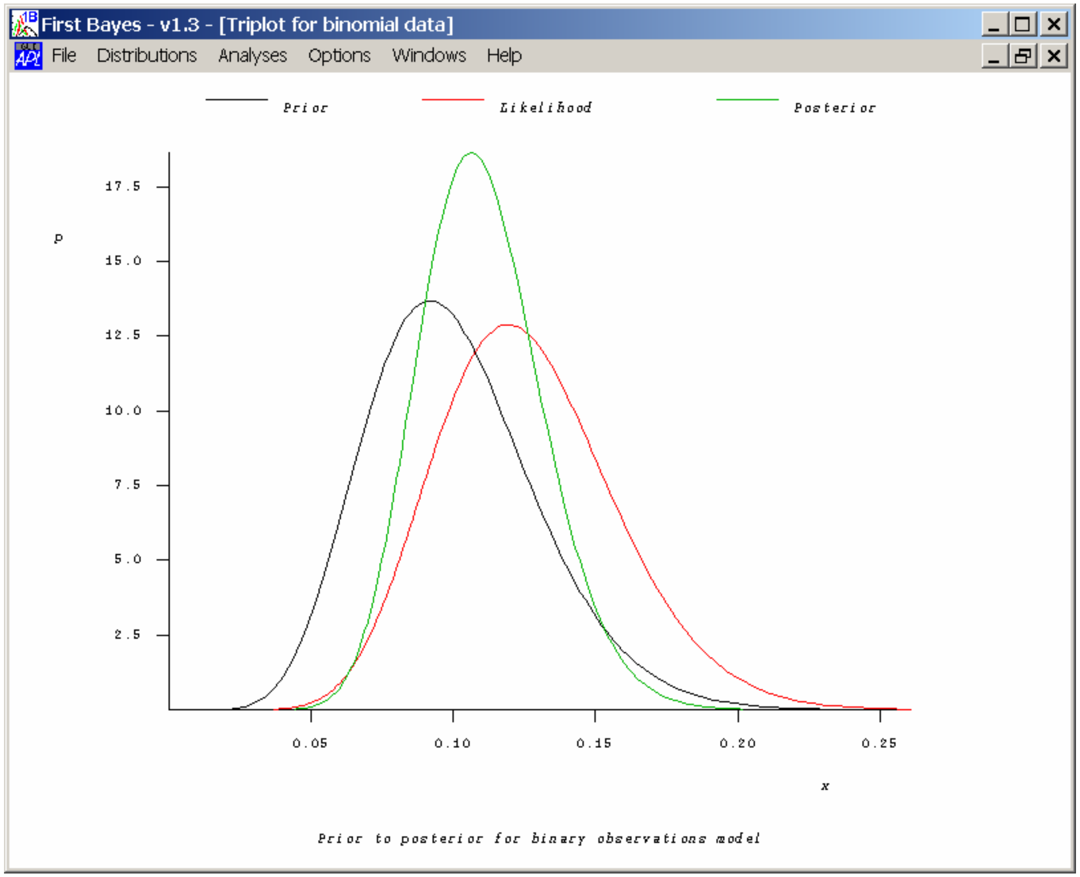
$$f(x \mid \theta) = \binom{n}{x} \theta^{x} (1-\theta)^{n-x}$$

- We obtain data that shows 2 failures in 5 trials

Prior Probability	Posterior Probability
P{θ=0.3}=0.1	P{θ=0.3}=0.13
P{θ=0.6}=0.9	P{θ=0.6}=0.87

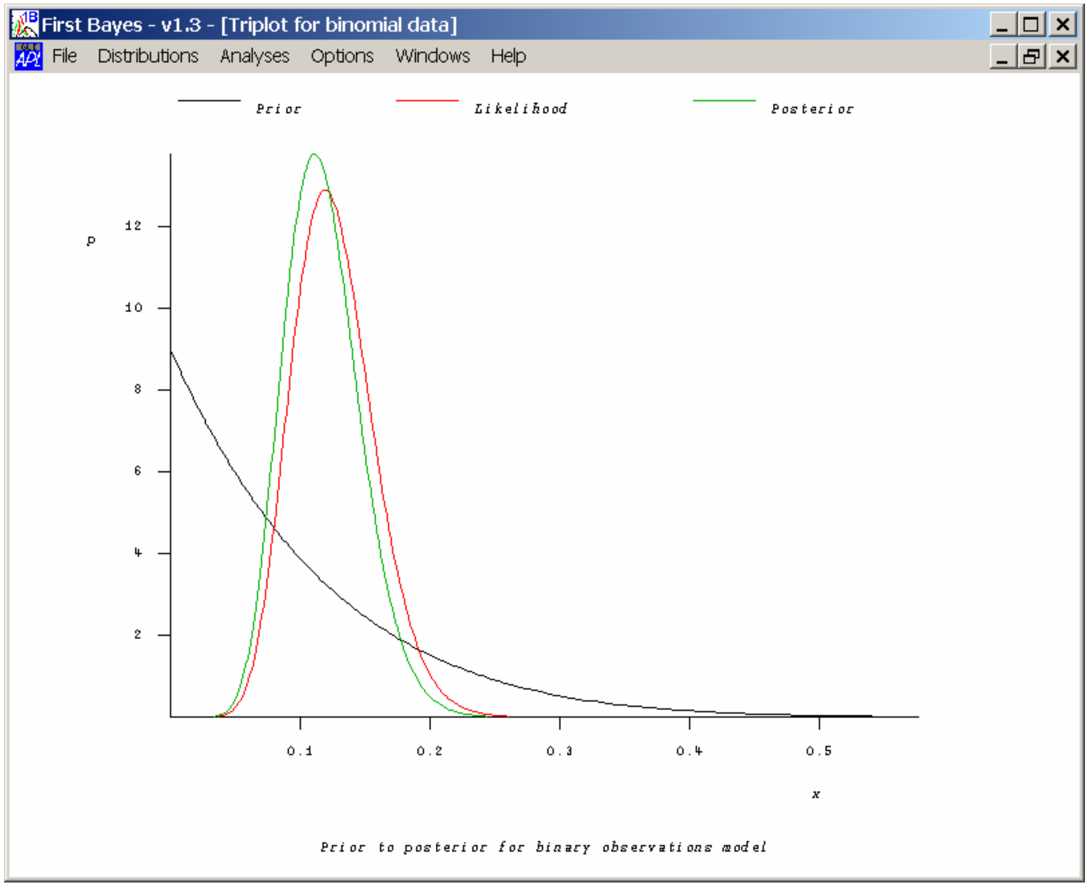
- The posterior distribution reflects the fact that in this set of data, θ = 0.4 which is closer to 0.3 than 0.6 and so the probability of θ=0.3 has risen slightly.

FirstBayes Software



The dataset is a string of ones and zeros, representing the failure or success of the Rosenbrock function, where failure is defined as a function value > 1000 over the input range $-2 \le x_1, x_2 \le 2$. Approximately 10% of the points “fail” according to this threshold.

FirstBayes Software


First Bayes - v1.3 - [Triplot for binomial data]
File
Distributions
Analyses
Options
Windows
Help
Prior
Likelihood
Posterior
p
12
10
8
6
4
2
0.1
0.2
0.3
0.4
0.5
x
Prior to posterior for binary observations model

If instead we take a prior that is “non-informative” (but still has a mean of 0.1), the prior has a much larger variance and so doesn’t influence the posterior as much. Notice that the posterior is much closer to the likelihood function.

How are posterior distributions calculated?

- In the case of conjugate pairs, one can analytically calculate the posterior distribution
- Most cases are too difficult to calculate analytically, thus we need to go to a sampling method
- Most popular approach is called Markov Chain Monte Carlo (MCMC)
- In MCMC, the idea is to generate a sampling density that is approximately equal to the posterior. We want the sampling density to be the stationary distribution of a Markov chain.

Markov Chain Monte Carlo

- How do we generate the Markov chain with the stationary probability that we want?
- Construct a transition probability that will get you there
- Metropolis-Hastings and Gibbs sampling are the most commonly used algorithms
- Both have the idea of a “proposal density” which is used for generating X_{i+1} in the sequence, conditional on $X_{i.}$ The proposal density is often denoted as $Q_Y(Y|X_i)$

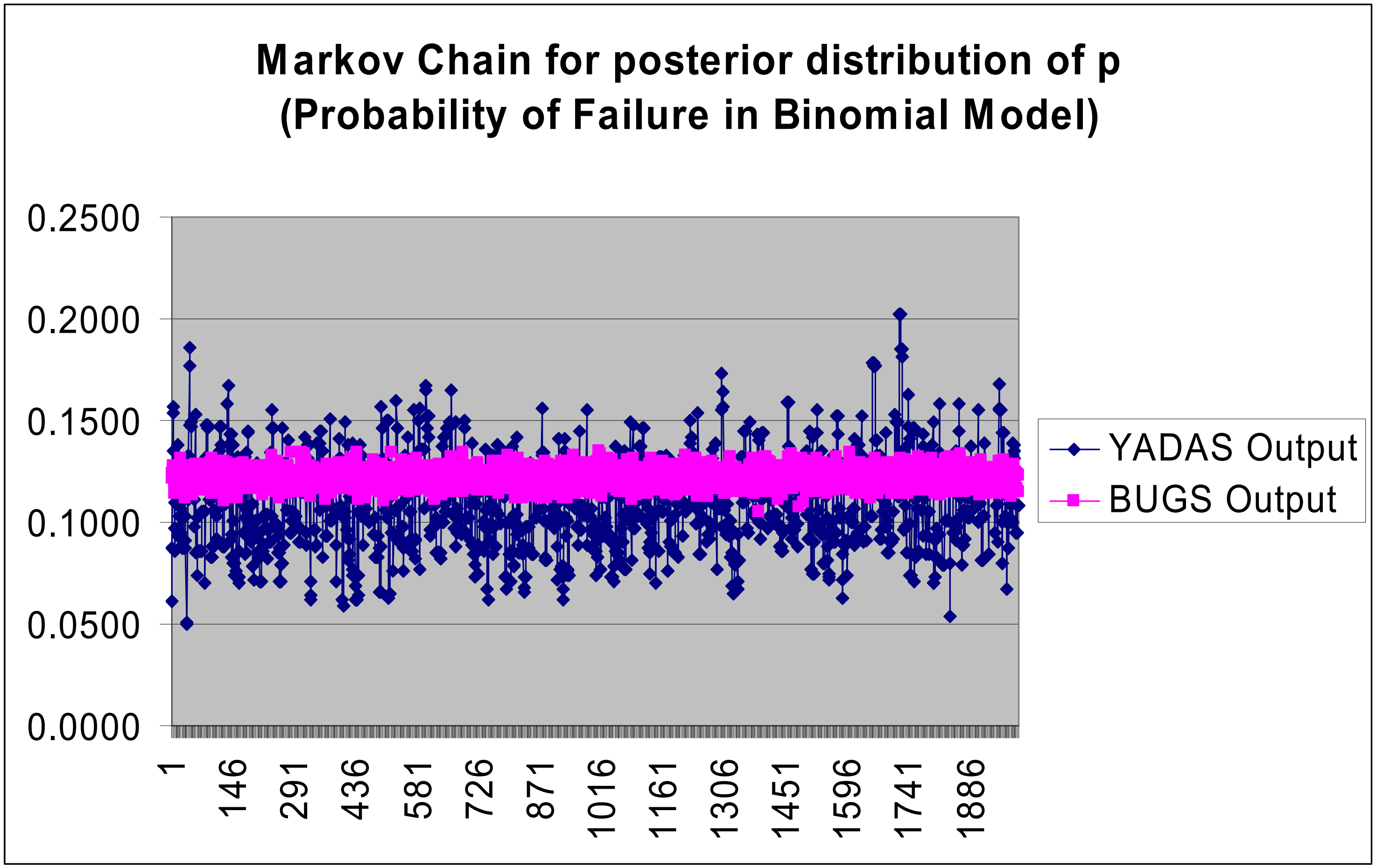
Metropolis-Hasting

- Basic method: generate a proposed sample from Q, calculate acceptance rate, calculate random number to see if candidate is accepted

$$\alpha(X,Y) = \min(1, \frac{f_X(Y) q_Y(Y \mid X_i)}{f_X(X) q_X(X_i \mid Y)})$$

- **Issues:**
  - Does Q, the proposal density, need a special form?
    - Symmetric *Q*(X|Y)=*Q*(Y|X).
    - Independent *Q*(Y|X)=*Q*(Y|)
  - How long do you run the chain, how do you know when it is converged, how long is the burn-in period, etc.?
  - ACCEPTANCE RATE is CRITICAL. Need to tune Q to get an “optimal” acceptance rate, 45-50% for 1-D problems, 23-26% for high dimensional problems
  - COMPUTATIONALLY VERY EXPENSIVE!!!!!!

BUGS and YADAS: Posterior distribution



Observations about MCMC

- It works best if your prior is well-defined and close to the posterior.
- It is very difficult to tell if the chain has converged to the "true" underlying posterior
- It requires substantial statistical knowledge to formulate the posterior "proposal" distribution correctly
- Each problem requires tuning of the parameters that govern the Markov chain generation – step sizes, "leaping" parameters, etc.

Some concerns about Bayes

- The Bayesian framework allows one to integrate observed data and prior knowledge: conceptually very nice.
- It won’t work well in cases where there is very little data or lots of data: optimum is where we have some data that is likely to be added to over time.
- In the context of many CS&E problems, we need to seriously question the usefulness of the Bayesian approach.

Example CS&E Bayesian Applications

- Estimation of probability of failure
- Estimation of hyperparameters that govern a surrogate model in a trust region or over the entire surface
    - **Experience with a linear regression model:**

$$E[y_i \mid \boldsymbol{\beta}, \mathbf{X}] = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik}$$

    - **Bayesian estimates for mean of β and for σ^2 are the same as those obtained by classical regression or by Maximum likelihood estimates**
  - **What does the Bayesian framework buy us? Are we really going to sample from values of the posterior of β to use in a simulation?**
- Multi-level surrogates
    - **Can we construct a surrogate based on a few high-resolution function evaluations, then update it with many low-resolution function evaluations or vice-versa? This is a promising area.**

Sandia National Laboratories

Bayesian Regression

- The parameter vector we are trying to estimate is:

  $\theta = (\beta,\sigma^2) = (\beta_0, \beta_1, \ldots , \beta_k, \sigma^2)$.

- The key assumption in a Bayesian formulation of regression is that there is a distribution on θ, and that the posterior distribution of θ is given by:

  $p(\theta|X,y) \propto p(\theta)p(y|X, \theta)$

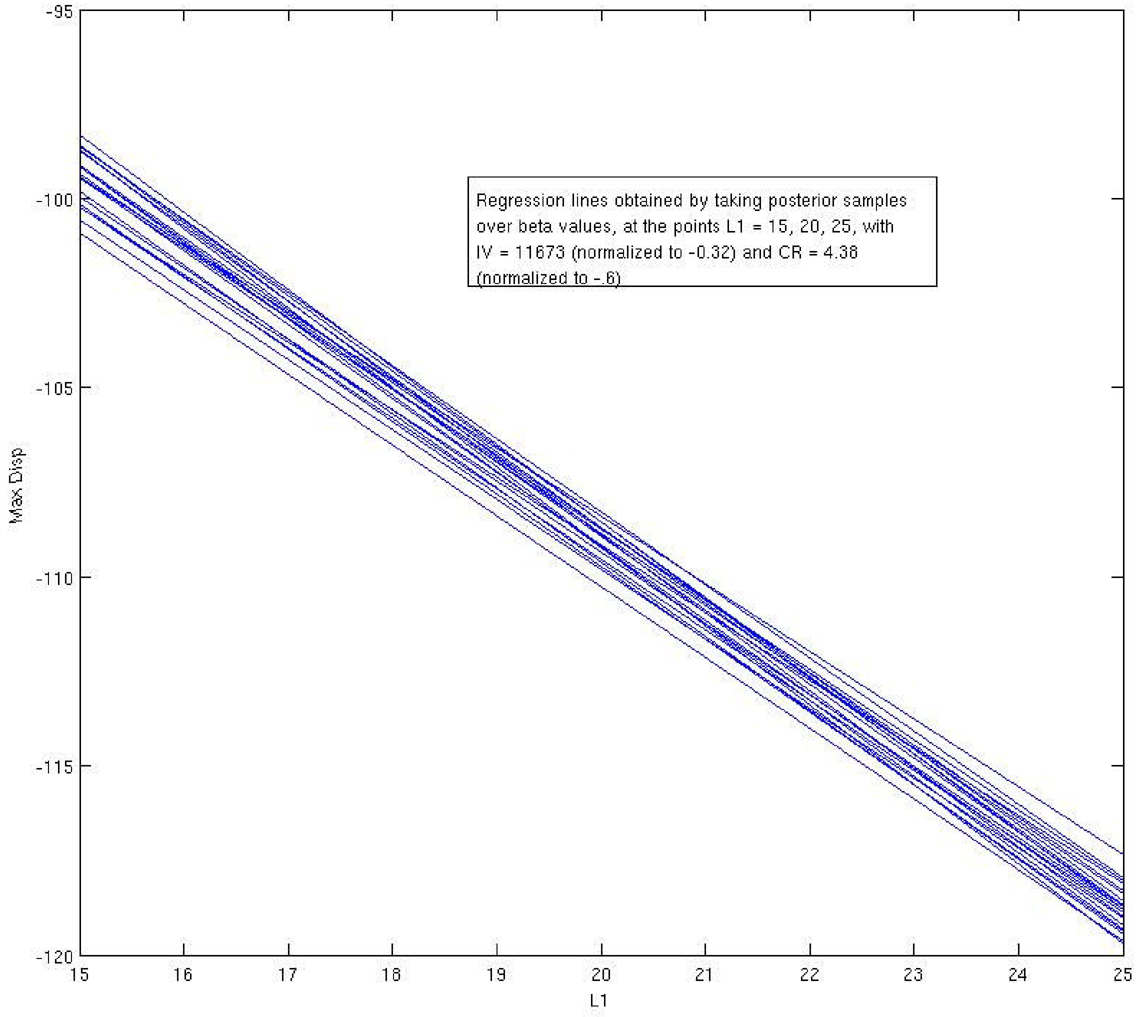
- With a noninformative prior θ, the conditional posterior of β given σ^2 is normal:

  $$\beta \,|\, \sigma^2, Y \sim N(\hat{\beta}, V_\beta \sigma^2)$$

- The marginal posterior density of σ^2 given the data is an inverse χ^2 distribution: $\sigma^2 \mid y \sim$ inverse χ^2 (n-k, s^2)

- The predicted outcome ŷ can be drawn directly from a multivariate-t, with center $\tilde{X}\hat{\beta}$, squared scale matrix $(I + \tilde{X}V_\beta \tilde{X}')s^2$, and (n-k) degrees of freedom.

Bayesian Regression Example



Regression lines obtained by taking posterior samples over beta values, at the points L1 = 15, 20, 25, with IV = 11673 (normalized to -0.32) and CR = 4.38 (normalized to -.6)

Bayesian Regression Example

```
betabest =
  -83.7303
   -1.9199
  -14.6057
  -13.7867
betapost =
  -84.9806   -1.8589  -15.0115  -13.6536
  -81.5903   -2.0193  -14.6206  -14.0590
  -86.1574   -1.7978  -14.3796  -13.8155
  -87.1254   -1.7450  -14.4879  -13.5610
  -83.4898   -1.9278  -14.9434  -13.9248
  -84.3119   -1.8625  -13.6025  -14.6832
  -83.0762   -1.9515  -14.5526  -13.6314
  -85.4060   -1.8570  -14.4533  -13.6969
  -84.1351   -1.9042  -14.1771  -14.0882
  -84.0462   -1.8918  -14.4630  -13.5398
  -83.8104   -1.9166  -14.8050  -13.8323
  -82.2354   -2.0130  -14.7307  -14.2078
  -82.4147   -1.9722  -14.1570  -14.2643
```

Bayesian Regression

- **Allows one to calculate families of linear response models, and generate predictions based on the posterior density (which incorporates the error in the regression PLUS the uncertainty in the parameters).**
- **Can be used in uncertainty analysis, ensemble calculation, applications where we want to calculate threshold probabilities**
- **Can be used in surrogate modeling**
- **Pros: Formulation allows for an analytic solution to the regression parameter posterior distributions.**
- **Our next research steps:**
  - Look at the use of Bayesian regression in multi-fidelity surrogate modeling. Issue: Need to have “problem matching” between the low and high fidelity
  - Maximize posterior density with respect the design variables of being in some region A:

    $$\text{posterior}(x) = P(y \in A | x, \text{data})$$
  - Can do this without Bayesian approach, but have more accurate representation of uncertainty with the posterior density

Calibration under Uncertainty

- **Idea: Want to account for both experimental uncertainty AND model uncertainty in the determination of model parameter values**
- Building on the work of Kennedy and O’Hagan.
- Formulate a relationship between observations, “true” process, and model output as: $z_i = \zeta(x_i) + e_i = \rho\ \eta(x_i,t_i) + \delta(x_i) + e_i$

  z is the observed data,

  t is the observed value of parameters θ,

  e_i is the observation error for the ith observation,

  ρ is an unknown regression parameter,

  $\delta(x)$ is a model discrepancy or model inadequacy function

- $\eta(x_i,t_i)$ and $\delta(x_i)$ are Gaussian process models. They are distributed with a mean and variance which are functions: e.g., $\eta(x_i,t_i) \sim N\ (h(x)^T\beta_{,}\ c(x, x'))$ where the covariance is often given as:

$$\mathbf{c(x - x')} = \sigma^2 \exp\{\sum_{j=1}^{q} \omega_j (x_j - x'_j)^2\}$$

Calibration Under Uncertainty

- Calibration involves calculating a very complicated joint pdf on all of these parameters: ρ, σ, ω, h terms, β, λ, and θ.
- Approach is to fix some of these terms, and estimate others. Even KOH admit at this point, this is not readily tractable.
- The “updating” does not have to be Bayesian – one could use Maximum likelihood as Dennis Cox at Rice does. This removes problems with generation of the posterior distribution.
- Whole approach is HIGHLY parameterized.

Model Discrepancy Term

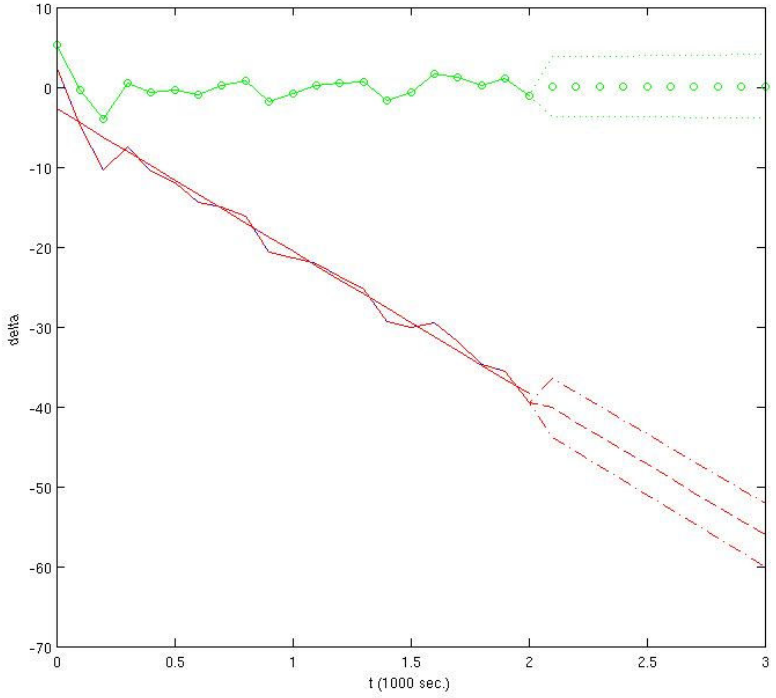
- Most useful concept in KOH work is the representation of the model discrepancy term as a random field
- Instead of constructing 2 emulators, just use one GP:

  KOH: $z_i = \rho\ \eta(x_i,t_i) + \delta(x_i) + e_i$

  SNL: $z_i = \text{CodeRuns} + \delta(x_i) + e_i$
- If we represent δ as a GP, how do we update the hyperparameters governing it? I am using MLE, not a Bayesian approach.
- MCMC is difficult to implement for multivariate cases and computationally expensive (because you need to do function evaluations in the acceptance or rejection of the proposed posterior distributions)
- If we replace the emulator with the actual code runs, we give up some flexibility and possibly calibration potential. However, it offers the possibility of being able to calibrate the code parameters directly → MY NEXT STEP

CUU Example

- Heat conduction example, want to predict temperature as a function of time
- Delta (in degrees) is a strong function of time
- GP mean (shown in green) was “corrected” by subtracting the regression term shown in red
- In this case, the GP updating doesn’t buy you much, since the GP reverts back to a zero mean, constant variance process
- Next steps: Looking at making the delta term a function of the model parameters



Conclusions

- Bayesian methods have a place in CS&E problems
- Need to be careful about what you claim as "Bayesian updating"
- Most natural applications are probability of failure estimation, Bayesian regression, and updating surrogate model parameters
- Major difficulty is generating a posterior distribution
- Even if one can develop a reasonable MCMC model, it will require a lot of function evaluations to generate the posterior → too computationally expensive for high-fidelity models